

REMARKS

Claims 1-4, 7-9 and 11-14 are now pending in the present application. Claims 1, 7-9 and 11-13 have been amended, Claim 10 has been canceled, and Claim 14 has been added, herewith. Reconsideration of the claims is respectfully requested.

I. 35 U.S.C. § 112, Second Paragraph

The Examiner rejected Claims 1-4 and 7-13 under 35 U.S.C. § 112, second paragraph, as being indefinite for failing to particularly point out and distinctly claim the subject matter, which applicants regard as the invention. This rejection is respectfully traversed.

With respect to Claims 1, 2, 7 and 8 – (sic), the Examiner states it is unclear what is meant by “strongly typed”. Applicants have amended Claims 1 and 8 to include the normal, ordinary meaning to those of ordinary skill in the art of the terminology “strongly typed”. Claim 2 depends upon Claim 1, and thus similarly includes clarification for this term. As to Claim 7, such claim has been amended to eliminate this terminology.

With respect to Claims 1 and 8, the Examiner states that the term ERP is not defined. Applicants have amended such claims to provide a definition for ERP per the present Specification. Applicants have also amended Claim 8 to provide proper antecedent basis for “an ERP application”.

With respect to Claims 9-11, the Examiner notes that such claims are directed to an article of manufacture, but depend upon respective method/system claims. Applicants have amended Claims 9 and 11 to be in independent form to overcome such rejection. Claim 10 has been cancelled herewith, without prejudice or disclaimer.

With respect to Claim 13, the Examiner notes that such claim is a method, and yet it depends upon a system claim. Applicants have amended Claim 13 accordingly.

Therefore the rejection of Claims 1-4 and 7-13 under 35 U.S.C. § 112, second paragraph has been overcome.

II. 35 U.S.C. § 102, Anticipation

The Examiner rejected Claims 8 and 13 under 35 U.S.C. § 102 as being anticipated by Beauchamp (6,621,505). This rejection is respectfully traversed.

For a prior art reference to anticipate in terms of 35 U.S.C. 102, every element of the claimed invention must be identically shown in a single reference. *In re Bond*, 910 F.2d 831, 15 USPQ2d 1566 (Fed. Cir. 1990). Applicants will now show that every element of the claimed invention is not identically shown in a single reference.

Specifically with respect to Claim 8 (and similarly for dependent Claim 13), Applicants show that the cited reference does not teach the claimed feature of “an ERP web gateway in communication with said web server for converting said requests from said web server into language format required by an interface of said ERP application and to convert information received from said interface of said ERP application into a strongly typed object form”. As can be seen, Claim 8 recites an ERP web gateway that performs two types of conversions – converting requests from a web server into language format required by an ERP application interface, and convert information received from such interface into a strongly typed object form. In rejecting Claim 8, the Examiner merely alleges that the cited reference teaches a gateway object *for translating* between the web server and the ERP database, cited Beauchamp’s fig. 7. Applicants show that a mere teaching of translating between a web server and ERP database does not teach the specific claimed step of converting information received from said interface of said ERP application into a strongly typed object form. Beauchamp merely states an ability to *access data* using a business object (Beauchamp Col. 20, lines 7-8; Col. 21, lines 44-48), but does not otherwise teach any ability to *convert this information into a strongly typed object*, as claimed. Thus, it is shown that every claimed element is not identically shown in a single reference, and Claim 8 (and similarly for dependent Claim 13) is therefore not anticipated by the cited reference.

This missing claimed feature advantageously allows for subsequent use of a template when converting this strongly typed object form to a transmittable format for display by a browser (Specification page 10, lines 4-12), thereby maintaining context of the data through such conversion and transmission (Specification page 19, line 24 – page 21, line 5).

Therefore, the rejection of Claims 8 and 13 under 35 U.S.C. § 102 has been overcome.

III. 35 U.S.C. § 103, Obviousness

A. The Examiner rejected Claims 1-4 and 9-10 under 35 U.S.C. § 103 as being unpatentable over the J2EE specification in view of Beauchamp (6,621,505). This rejection is respectfully traversed.

Applicants show that the cited references have been improperly combined using hindsight, and even when improperly combined there are still missing claimed elements – strongly evidencing non-obviousness. In combining the references, the Examiner states:

“it would have been obvious to a person of ordinary skill in the art at the time the invention was made to combine the teachings of Beauchamp with the J2EE specification because this would allow the access to legacy ERP applications as taught by Beauchamp to be added to the data sources already supported by the J2EE specification”.

Applicants show that this general statement – which essentially states that the references can be combined, thus providing a benefit – is not a sufficient showing of a motivation to combine these references. “[w]hen determining the patentability of a claimed invention which combines two known elements, ‘the question is whether there is something in the prior art as a whole to suggest the desirability, and thus the obviousness, of making the combination.’” *See In re Beattie*, 974 F.2d 1309, 1311-12, 24 USPQ2d 1040, 1042 (Fed. Cir. 1992) (quoting *Lindemann Maschinenfabrik GmbH v. American Hoist & Derrick Co.*, 730 F.2d 1452, 1462, 221 USPQ 481, 488 (Fed. Cir. 1984)). There is certainly nothing in the cited art that suggests the desirability of making such combination. In fact, there is something in the cited art that would suggest a *non-desire* to make the combination. The Beauchamp reference provides for supporting legacy ERP applications. However, the cited J2EE reference expressly states its tools do not enable support for such legacy ERP applications (J2EE page 2-2, last sentence), and that such support will not be available until some undefined future release (Applicants’ note: if this were an obvious extension, why the delay in implementation?). This strongly evidences no motivation to combine the references as the J2EE teachings do not operate to support legacy ERP applications as taught by Beauchamp. The only motivation to combine the

references comes from Applicants' own patent specification, which is improper hindsight analysis. It is error to reconstruct the patentee's claimed invention from the prior art by using the patentee's claims as a "blueprint". When prior art references require selective combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight obtained from the invention itself. *Interconnect Planning Corp. v. Feil*, 774 F.2d 1132, 227 USPQ 543 (Fed. Cir. 1985). There is simply no reason to combine the references other than the hindsight obtained by the present invention.

Even when improperly combined, there is still at least one missing claimed element not taught or suggested by the cited references. In particular, none of the cited references teach or suggest the claimed step of "merging said output data from said ERP application into a strongly typed object". As can be seen, Claim 11 expressly requires that output data from an ERP application be *merged into* a strongly typed object. The passages cited by the Examiner do not teach or suggest any type of merging step or function. The passage cited by the Examiner at Beauchamp Col. 21, lines 59-60 states:

"In one embodiment, business objects may be named Java classes that provide access to any external data required"

As can be seen, this passage merely states use of objects to access data. Claim 1 goes further, and expressly recites the merging of data into a strongly typed object. Providing access to data using an object is very different from the specific claimed step of merging data *into* a strongly typed object. Thus, it is shown that this cited passage does not teach or suggest the claimed merging step recited in Claim 1.

Nor does the passage cited by the Examiner at Beauchamp Col. 22, lines 5-10 overcome this deficiency. There, Beauchamp states:

"The data access administrator first creates a repository data definition (RDD) that describes the business object data, as indicated at 270. A RDD is a definition of the data elements that are contained in a business object. In a preferred embodiment, the format of the RDD is XML, and it

specifies at a field level the BO name, data type, and other attributes that are necessary to describe the BO".

This passage merely describes creation of a description of object data using XML. These objects are manually created and maintained by a data base administrator (Beauchamp Col. 22, lines 1-2). There is no hint, teaching or suggestion that the data contained therein comes from an ERP application and has been merged into the object, as claimed. While these objects encapsulate specific calls to retrieve and convert data (Beauchamp Col. 22, lines 44-45), there is no teaching or suggestion that this data *is merged into the object itself*, as claimed. Therefore, the Examiner has not established a prima facie case of obviousness with respect to Claim 1¹, and the burden has therefore not shifted to Applicants to rebut an obviousness assertion². In addition, as the Examiner has failed to establish a prima facie showing of obviousness, the rejection of Claim 1 is shown to be improper³.

Applicants traverse the rejection of Claims 2-4 for reasons given above regarding Claim 1 (of which Claims 2-4 depend upon).

Applicants traverse the rejection of Claim 9 for similar reasons to those given above regarding Claim 1.

Therefore, the rejection of Claims 1-4 and 9 under 35 U.S.C. § 103 has been overcome.

B. The Examiner rejected Claims 7 and 11 under 35 U.S.C. § 103 as being unpatentable over XMLC Tutorial. This rejection is respectfully traversed.

¹ In rejecting claims under 35 U.S.C. Section 103, the examiner bears the initial burden of presenting a prima facie case of obviousness. *In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992). To establish prima facie obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03. See also, *In re Royka*, 490 F.2d 580 (C.C.P.A. 1974).

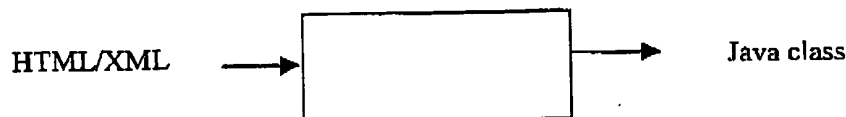
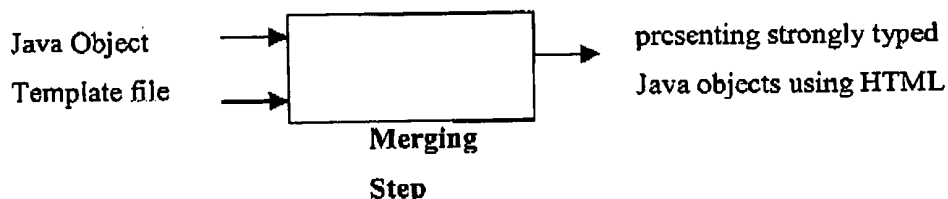
² Only if that burden is met, does the burden of coming forward with evidence or argument shift to the applicant. *In re Oetiker* supra.

³ If the examiner fails to establish a prima facie case, the rejection is improper and will be overturned. *In re Fine*, 837 F.2d 1071, 1074, 5 USPQ2d 1596, 1598 (Fed. Cir. 1988).

With respect to Claim 7 (and similarly for Claim 11), such claim recites a method of presenting strongly typed Java objects using HTML “by merging Java objects with XML template files”. As can be seen, Claim 7 expressly recites a merging step, and the two items expressly recited as a part of the merging step are (i) Java objects, and (ii) XML template files. Thus, Claim 7 expressly recites merging of a Java object with an XML template file. The cited reference does not teach or suggest any type of merging step, and specifically does not teach or suggest the merging of Java objects with XML template files. In rejecting Claim 7, the Examiner states that the XMLC tutorial teaches a method of presenting Java objects using HTML by merging Java objects with XML template files at page 3, lines 6-9. Applicants show that there, the cited XMLC tutorial states:

“XMLC is a Java-based compiler that takes a document written in Hypertext Markup Language (HTML) or the eXtensible Markup Language (XML) and *creates Java classes* that will faithfully recreate the document. The *resulting Java classes* can be used to insert dynamic content into a document framework at runtime. XMLC, therefore, is a wonderful way to create dynamic HTML or XML documents from Java” (emphasis added by Applicants).

As can be seen, this passage merely describes the *creation* of a Java class *from* an HTML or XML document. This is different from the invention as recited in Claim 7, as can easily be seen by the following graphical depiction:

XMLC Tutorial teachingClaim 7 claimed feature

As can be seen from the above, there is no merging of two items in the teachings of the cited reference, and in particular there is no teaching or suggestion of merging Java objects with XML template files.

In addition, Claim 7 expressly recites a method for presenting Java objects using HTML. The resulting Java class as taught by the cited reference is not viewable or presentable (note in particular the last line on page 4 of the cited reference, where it states "The command shown would generate a Java class file called "hello.class" in the same directory. Note that this is compiled Java bytecode; *you would not be able to read it*" (emphasis added by Applicant)). Thus, the cited passage provides no ability or method of *presenting* Java objects, as claimed.

In fact, the teachings of the cited reference are, in effect, doing just the opposite of what is claimed in Claim 7. The cited reference teaches converting HTML into a Java class. Applicants have amended Claim 7 (and similarly for Claim 11) to emphasize that the teachings of the cited reference are diametric to what is being claimed in Claim 7. In particular, amended Claim 7 recites "A method of presenting Java objects using HTML by merging the Java objects with XML template files to generate the HTML used to present the Java objects". As can be seen, the claim recites generation of HTML from Java objects (which are merged with XML template files), whereas the cited passage

teaches generation of Java classes from HTML. These are diametric concepts. As every element of the claimed invention is not taught or suggested by the cited reference, it is shown that Claim 7 is not obvious in view of the cited reference¹. Therefore, the rejection of Claim 7 (and similarly for Claim 11) has been overcome.

C. The Examiner rejected Claim 12 under 35 U.S.C. § 103 as being unpatentable over XMLC Tutorial in view of Beauchamp (6,621,505).

Applicants initially traverse the rejection of Claim 12 for reasons given above regarding Claim 7 (of which Claim 12 depends upon), and show that none of the cited references teach or suggest a method for (i) presenting Java objects, nor do they teach or suggest a step of (ii) merging Java objects with XML template files to generate the HTML used to present the Java objects.

Further with respect to Claim 12, Applicants show that none of the cited references teach or suggest the claimed step of "merging output data from an ERP application into at least one of the Java objects" as a part of the overall method of presenting a Java object. Reasons why this merging step is missing in the cited Beauchamp reference have been previously described above with respect to Claim 1, and such reasoning similarly applies for Claim 12. Therefore, it is shown that the Examiner has not established a *prima facie* case of obviousness with respect to Claim 12, and the burden has therefore not shifted to Applicants to rebut an obviousness assertion. In addition, as the Examiner has failed to establish a *prima facie* showing of obviousness, the rejection of Claim 12 is shown to be improper.

Applicants also show that the Examiner has failed to establish a proper motivation for combining the references. The reason given by the Examiner is combining the references is "to allow the architecture of the XMLC Tutorial to be used to access ERP data sources". In effect, the Examiner's motivation to combine the references is that they can be combined, to be used together. Such a broad-brushed rationale is clearly contrary

¹ To establish *prima facie* obviousness of a claimed invention, all of the claim limitations must be taught or suggested by the prior art. MPEP 2143.03. *See also, In re Royka*, 490 F.2d 580 (C.C.P.A. 1974) (emphasis added by Applicants). In the absence of a proper *prima facie* case of obviousness, an applicant who complies with the other statutory requirements is entitled to a patent. *See In re Oetiker*, 977 F.2d 1443, 1445, 24 USPQ2d 1443, 1444 (Fed. Cir. 1992).

to current case law. As stated by the Federal Circuit, "virtually all [inventions] are combinations of old elements." *Environmental Designs, Ltd. v. Union Oil Co.*, 713 F.2d 693, 698, 218 USPQ 865, 870 (Fed. Cir. 1983); *see also Richdel, Inc. v. Sunspool Corp.*, 714 F.2d 1573, 1579-80, 219 USPQ 8, 12 (Fed. Cir. 1983) ("Most, if not all, inventions are combinations and mostly of old elements."). Therefore an examiner may often find every element of a claimed invention in the prior art. If identification of each claimed element in the prior art were sufficient to negate patentability, very few patents would ever issue. Furthermore, rejecting patents solely by finding prior art corollaries for the claimed elements would permit an examiner to use the claimed invention itself as a blueprint for piecing together elements in the prior art to defeat the patentability of the claimed invention. Such an approach would be "an illogical and inappropriate process by which to determine patentability." *Sensonics, Inc. v. Aerosonic Corp.*, 81 F.3d 1566, 1570, 38 USPQ2d 1551, 1554 (Fed. Cir. 1996). To prevent the use of hindsight based on the invention to defeat patentability of the invention, this court requires the examiner to show a motivation to combine the references that create the case of obviousness. **In other words, the examiner must show reasons that the skilled artisan, confronted with the same problems as the inventor and with no knowledge of the claimed invention, would select the elements from the cited prior art references for combination in the manner claimed.** *In re Rouffet*, 149 F.3d 1350, 47 USPQ 2d 1453 (Fed. Cir. 1998) (emphasis added by Applicants). The Examiner has failed to meet this burden (as the Examiner merely alleges that the two teachings can be used together), and thus the references have been combined without providing proper motivation for the combination. Thus, Claim 12 is further shown to have been erroneously rejected.

Applicants further show that the references have been improperly combined using hindsight analysis¹. The cited XMLC tutorial describes converting HTML documents into Java classes and has nothing to do with accessing ERP data sources. The only motivation to modify the teachings of the XMLC tutorial to include data from an ERP application comes from Applicants' own patent application, which is improper hindsight

¹ It is error to reconstruct the patentee's claimed invention from the prior art by using the patentee's claims as a "blueprint". When prior art references require selective combination to render obvious a subsequent invention, there must be some reason for the combination other than the hindsight obtained from the invention itself. *Interconnect Planning Corp. v. Feil*, *supra*.

analysis. Thus, Claim 12 is still further shown to have been erroneously rejected by the Examiner.

Therefore, the rejection of Claim 12 under 35 U.S.C. § 103 has been overcome.

IV. Newly added Claim

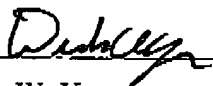
Claim 14 has been added herewith. Specification support for such claim is shown to be at least at Specification page 21, line 6 – page 26, line 15. Examination of such claim is respectfully requested.

V. Conclusion

It is respectfully urged that the subject application is patentable over the cited references and is now in condition for allowance. The Examiner is invited to call the undersigned at the below-listed telephone number if in the opinion of the Examiner such a telephone conference would expedite or aid the prosecution and examination of this application.

DATE: 6/28/04

Respectfully submitted,



Duke W. Yee
Reg. No. 34,285
Wayne P. Bailey
Reg. No. 34,289
Yee & Associates, P.C.
P.O. Box 802333
Dallas, TX 75380
(972) 367-2001
Attorneys for Applicants